SAP NetWeaver Process Integration 7.1
Advanced Adapter Engine

SAP NetWeaver Regional Implementation Group
SAP NetWeaver Product Management
December 2007

THE BEST-RUN BUSINESSES RUN SAP™

SAP

Benefits

SAP

After reading this presentation you will be able to:

- Explain the architecture of the Advanced Adapter Engine (AAE)
- Describe the benefits, features and configuration of the AAE
- Identify valid integration scenarios empowered by the AAE

Agenda

1. **Introduction and Architecture**
2. **Features and Configuration**
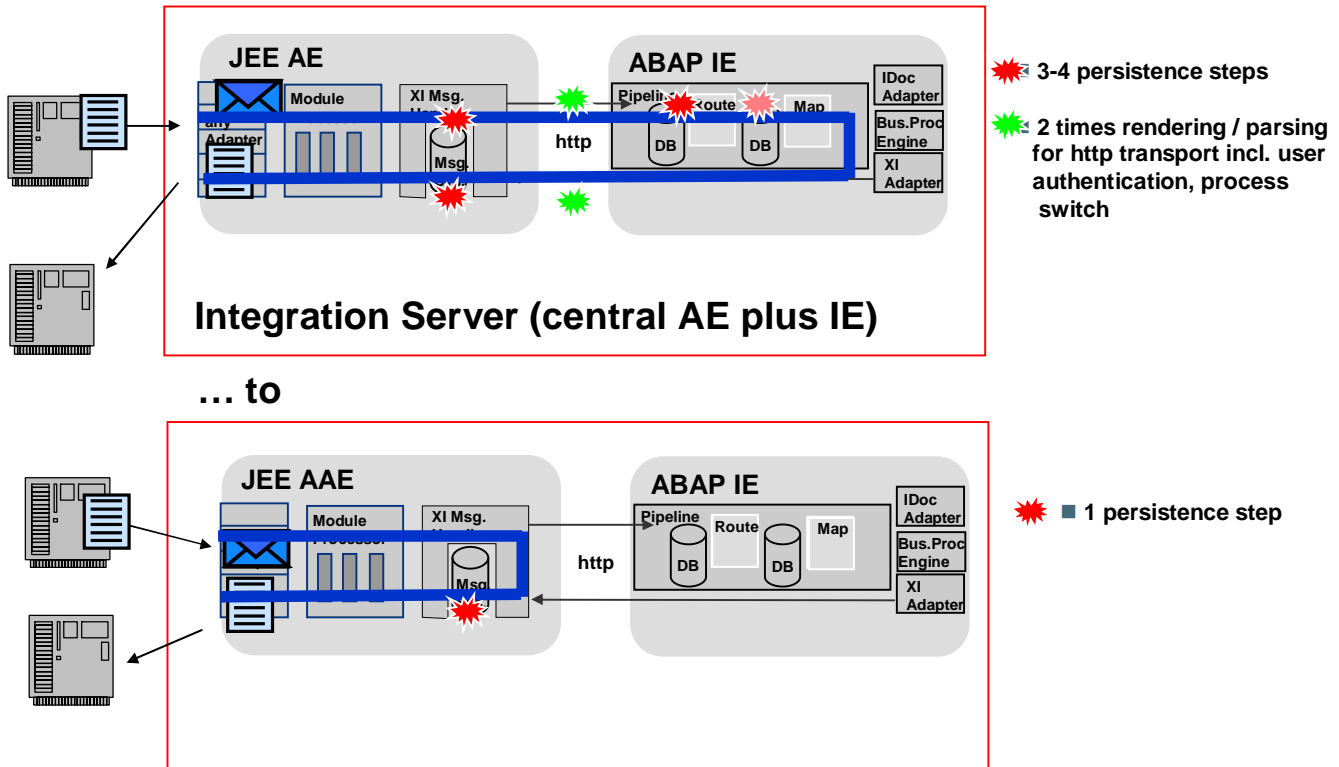3. **Integration Scenarios**
4. **Summary**

Agenda

1. **Introduction and Architecture**
2. **Features and Configuration**
3. **Integration Scenarios**
4. **Summary**

From Adapter & Integration Engine to Advanced AE (AAE)

Runtime Picture: From…

JEE AE — **ABAP IE**

Module | XI Msg. | Pipeline | Route | Map | IDoc Adapter | Bus.Proc Engine | XI Adapter

any Adapter | Msg. | http | DB | DB

- 3-4 persistence steps
- 2 times rendering / parsing for http transport incl. user authentication, process switch

Integration Server (central AE plus IE)

… to

JEE AAE — **ABAP IE**

Module Processor | XI Msg. | Pipeline | Route | Map | IDoc Adapter | Bus.Proc Engine | XI Adapter
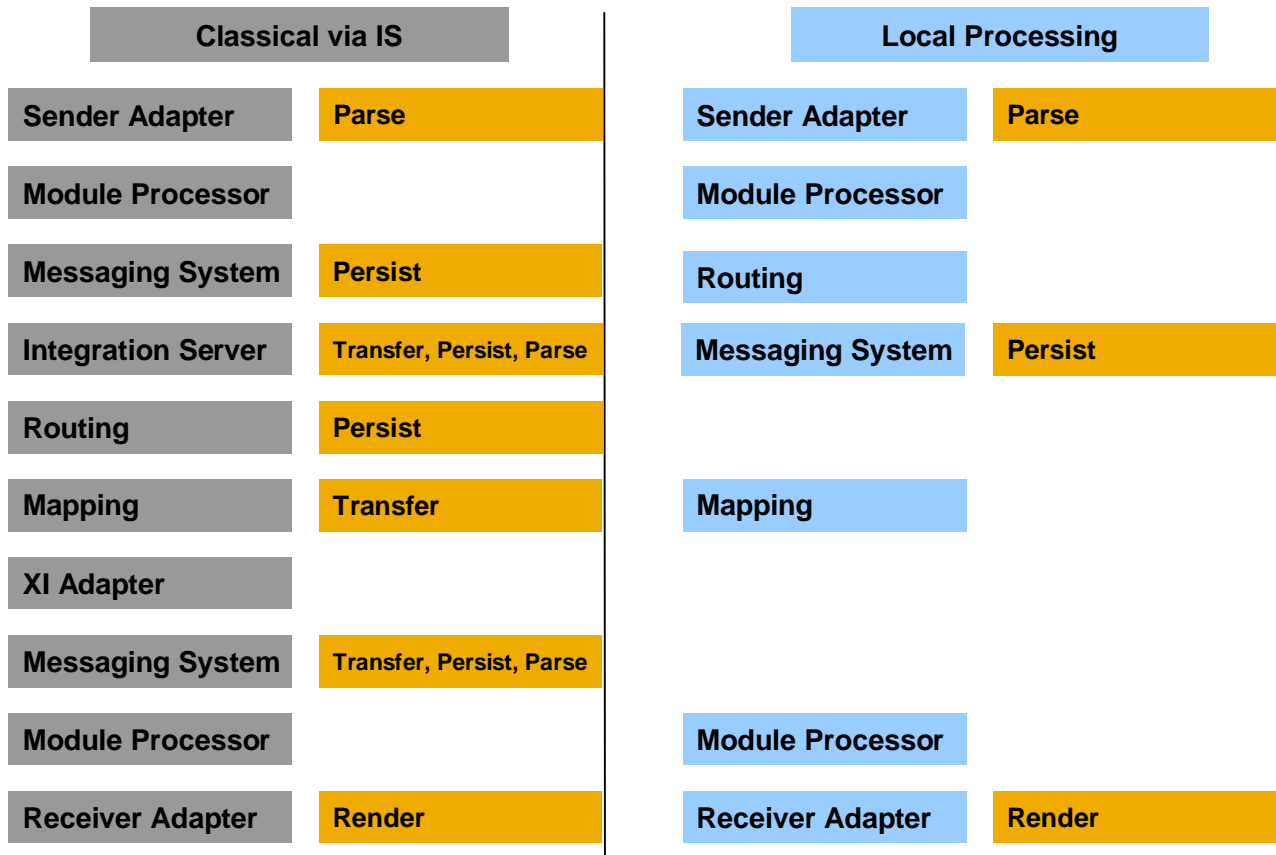
Msg | http | DB | DB

- 1 persistence step

- The Advanced Adapter Engine (AAE) is an evolution of the Adapter Engine, that is known from former releases. It is part of SAP NW PI 7.1 and from a deployments perspective, there is just the deployment for an AAE (i.e. there is no deployment option for the AAE OR AE!!). It is just up to the configuration if one uses the advanced features for a particular scenario, but not an installation/deployment decision! It runs on SAP NW JEE5.

- Objectives of SAP NW PI 7.10 are High Volume Support and to be the cornerstone of customers' SOA strategy. The new AAE with its features targets both of them.

- The AAE enables for end-to-end message processing without any help from the Integration Engine during runtime.

- It increases message processing speed and throughput by reducing latencies and resource consumption (both memory and processing time).

- Local Processing in AAE: Initegration Server (IS) is not involved during runtime (mapping and routing is also carried out within the AAE)

- Classical Processing with AAE: Processing also through the IS

- Local processing is probably feasible for existing java-based adapters, and it represents a performance and resource usage improvement.

- The AAE is available centrally (i.e. with installation and instance of IS) and non-centrally.

Local vs. „Classical" Processing

Classical via IS		Local Processing	
Sender Adapter	Parse	Sender Adapter	Parse
Module Processor		Module Processor	
Messaging System	Persist	Routing	
Integration Server	Transfer, Persist, Parse	Messaging System	Persist
Routing	Persist		
Mapping	Transfer	Mapping	
XI Adapter			
Messaging System	Transfer, Persist, Parse		
Module Processor		Module Processor	
Receiver Adapter	Render	Receiver Adapter	Render

Individual Steps for classical processing via IS:

1. The message enters the Sender Adapter where it is parsed.

2. Then it is sent to the module processor.

3. Subsequently it goes to the messaging system, where it is persisted.

4. Later the message is transmitted to the Integration Engine

   There, the message is transformed into XML format, replicated, authenticated, parsed again, persisted on IS.

5. Then the message is logically routed to the destinations and persisted again.

6. For each destination the message is mapped (usually executing a JCO call to the java stack).

7. Then the XI adapter starts the transmission back to the Adapter Engine executing all the internal steps it did before.

8. After that the module processor takes action again.

9. Finally the Message is sent to the adapter and to the target system.

Having the new local processing configured, several steps can be skipped: less persistency steps, less communication steps (AE<->IS, Mapping)

SAP NW PI 7.1 Landscape
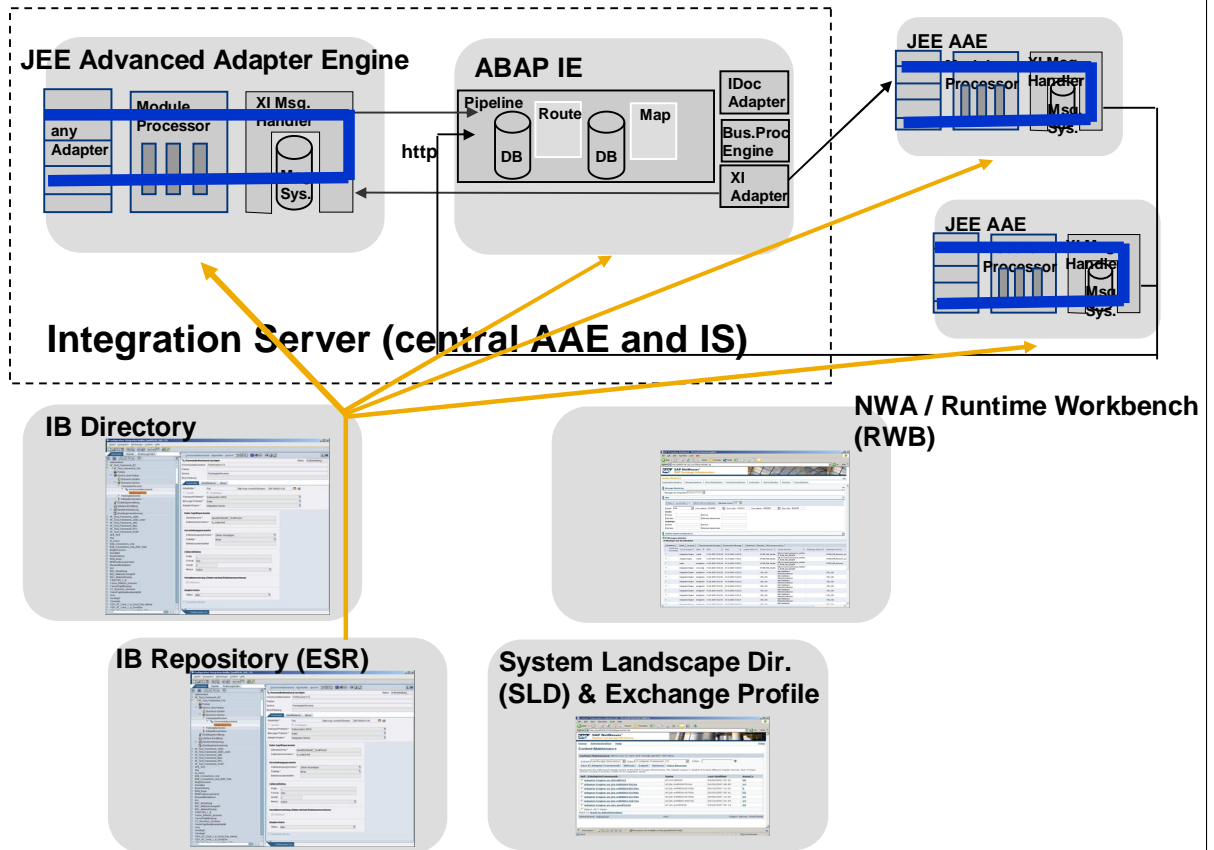
The central and non-central AAE is supported.

- On the central instance from a deployment view: each component/package is deployed just once (i.e. there is no messaging system for AAE and AE, no adapter type xyz for AE or AAE). There is just one monitoring per adapter type independent if it is used with the central IS or in a AAE-local scenario.

- From an adapter developments and validations process point of view there is also no difference, i.e. the adapter development needs no case distinction within its coding for local and classical processing.

- The decision if the JEE AAE is used solely or if the IS is involved, is a feature of the runtime, that takes a certain configuration into account.

- Besides the already existing cache mechanisms originating and well known from former PI-releases there is an additional cache mechanism involved for mappings.

- The known CPA cache is extended with objects resulting from the integrated configuration object, that also includes the routing information now and refers to the mapping that is separately distributed for AAE (formerly runtime cache, i.e. RWB => Cache Monitoring => „Integration Server Java Cache", containing mapping, value mapping and channels for lookup-API).

- PI 7.1 also still supports the configuration and runtime of PI 7.0 based non-central adapter engines via PI 7.1.

SAP

Local Processing in the Advanced Adapter Engine

- **Natural evolution of adapter engine**
- **Provides mapping, routing to by-pass Integration Server**
- **Adapter to adapter communication**
- **For sync and async scenarios**
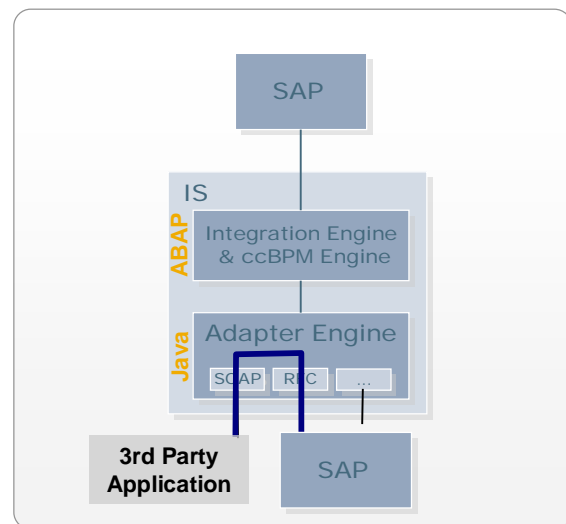
- Customers require high-volume throughput.

- The Adapter Engine (central and non-central) evolves to "local processing engine" -> non-disruptive evolution, can be used like a non-central Adapter Engine.

- It provides mapping, routing etc., and allows to by-pass the Integration Server and improves performance on the IS.

- Improved performance/message turnaround for adapter-to-adapter communication

- Central configuration and monitoring in Integration Directory and Runtime Workbench

- Allows customers to distribute execution of mission-critical processes

- Open for further (deployment) scenarios

- Usage of local processing within AAE and integrated configuration is optional, i.e. processing via IS is still another option

Agenda

1. Introduction and Architecture
2. **Features and Configuration**
3. Integration Scenarios
4. Summary

Advanced Adapter Engine – Features

SAP

- **Both synchronous and asynchronous messages**
- **Sender and receiver adapters located on the same Adapter Engine**
- **Support of decentral Adapter Engine with central configuration and monitoring**
- **Available for adapter types hosted on the Adapter Engine (IDoc-, http-, XI-, iSpeak- adapters excluded)**
- **One to one message mapping (no ABAP based mapping types)**
- **Single Static Routing**
  - **(i.e. no content based routing or message split yet)**

- The scope of the AAE:
  - Supports synchronous and asynchronous messages
  - Supports non-central Adapter Engines
  - And supported all java-based adapters already in use, in the future some other adapters will be supported as well.
- In the following service packages additional functionality is planned to be added for this same PI 7.1 version, for example it is intended to support:
  - IDoc and Web services
  - More routing and mapping functionalities
  - Additional optional persistence steps
- Plain-http can be implemented with SOAP-Adapter; XI adapter can be implemented technically on project basis with SOAP-Adapter in Axis-mode
- iSpeak: scenarios often use IDoc, ccBPM

Advanced Adapter Engine – Additional Available Features

- ✓ **ACL's**

- ✓ **Mapping Lookups**

- ✓ **Principal Propagation (if supported by the adapter)**

- ✓ **Value Mapping, Sequence Mapping**

- ✓ **Schema Validation**

- ✓ **Header Mapping (if supported by the adapter)**

- ✓ **Message Prioritization**

■ Async => Sync-bridge without BPM works.

Directory Configuration: Integrated Configuration

New Directory Object: Integrated Configuration Object

Used for configuring local processing within adavanced adapter engine. Therefore the AAE provides routing and mapping services locally.

Object Type used in Directory

ICO

IS-pipeline NOT invoked

IS-pipeline invoked
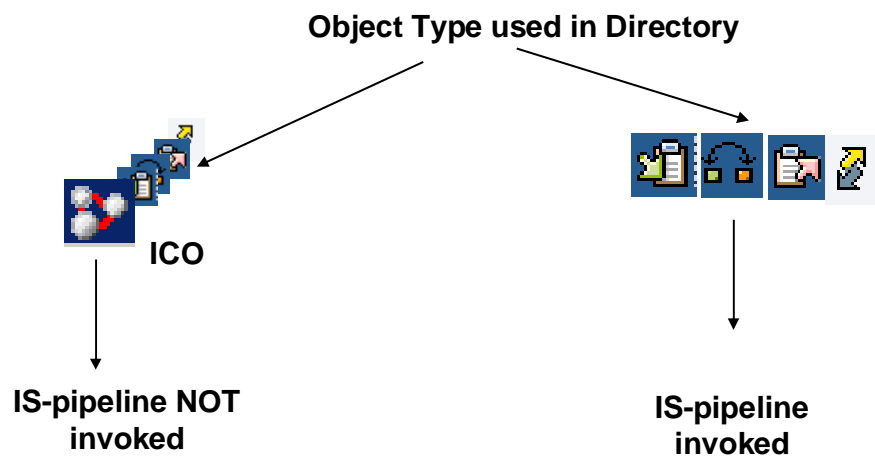
■ Local Processing is always optional.

Directory Configuration: Integrated Configuration

SAP

= ([icon] + [icon] + [icon] + [icon])

Integrated Configuration Object (ICO)

- **Contains sender- & receiver agreement and interface determination**
- **Is transferred to the CPA-Cache of the individual AAE with a reference to the mapping maintained within the interface determination section**
- **For each operation in interface determination a separate entry is created in CPA (receiver interface determination)**
- **The integrated configuration object is transferred as one single object to the AAE.**

Details – Directory Configuration

From four configuration entities to one single entity
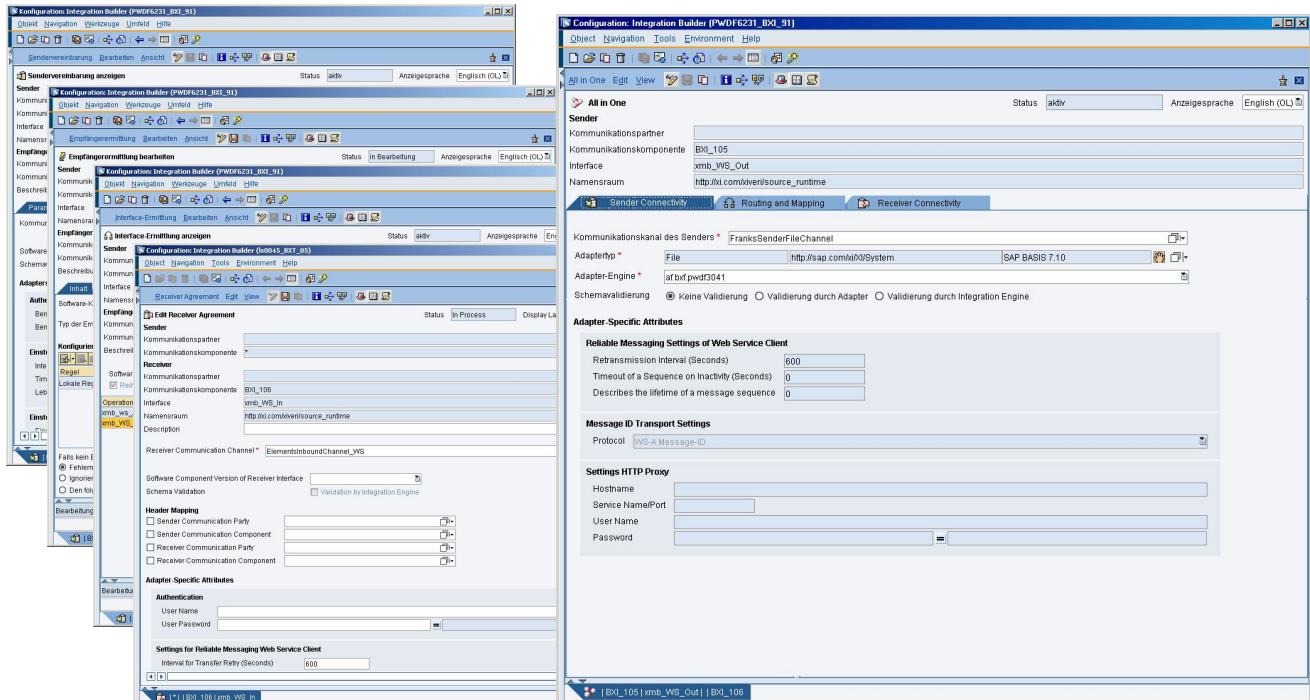
- At configuration time, instead of creating a Sender Agreement, Receiver Determination, Interface Determination and Receiver Agreement, you have to create an "Integrated Configuration" object.

- For routing, entering a receiver is optional (e.g. Virtual Receiver for Sender) can be used.

- The Integrated Configuration Object is used for local processing in an AAE only!

- ACL can be assigned by using a separate section in the Integrated Configuration Object

- The "Integrated Configuration" object, divides in three tab strips al the required configurations. They are:

  - The "Sender Connectivity":   Basically corresponds to the "Sender Agreement".

  - The "Routing and Mapping":   It comprises the "Receiver Determination" and "Interface Mapping" objects.

  - The "Receiver Connectivity: Basically corresponds to the "Receiver Agreement".

- Upon activation the "Integrated Configuration" is replicated to the AAEs local caches. That means that all the routings and mappings are executed locally.

Integrated Configuration Objects, Mapping & Cache

SAP

Delta Cache Refresh:

CPA-Cache and Mapping Runtime Cache are updated independently, i.e.

1. **Mappings objects per activation/change are transferred to ALL Mapping Runtime Caches of all Advanced Adapter Engines**

2. **CPA-Cache-Refresh is usually carried out after activation/change of directory objects. Objects are just transferred to the Advanced Adapter Engine where configured to be used**

Can be viewed in the Cache Monitoring of the RWB. (other monitoring is reused; no SXMB_MONI)

- *Initial Full Cache Refresh*: after the first start of the AAE, a flag is set for a full CPA-Cache refresh; it will be reset after exactly one *SUCCESSFUL* full cache refresh; full cache refresh can be always triggered from adapter engine later-on in addition

- Process and Results:

  CPA-Cache gets all known objects, configured and activated for the particular AAE, inclusively the new routings and mapping references(!) resulting from the Integrated Configuration Objects

- The new mapping runtime cache corresponds to the former „Integration Server Java Cache" (Java Runtime Cache) in previous releases but is now integral part of the adapter engine that also results in a decentral deployment option (Mapping Cache in Decentral AAE)

AAE

Advanced Adapter Engine

„classic AE"
(http to IS)

| Any Sender Adapter | Module Processor | XI Msg. Handler | Receiver Determination | | Interface Determination | | Module Processor | Any Receiver Adapter |

Check for ICO

Kept Compatible vs. AAE!
Channel, Agreement Sender

Integrated Conf. Obj.

Messag. System

Integrated Conf. Obj.

Mapping Runtime

Mapping Programs

Kept Compatible vs. AAE!
Channel, Agreement Receiver

Directory Configuration („CPA Cache")

Mapping Cache

Directory

Channels, Agreements

Integrated Configuration

Mapping programs

Legend:

New dev. In AAE

Enhanced For AAE

Flow of Runtime Data (messages)

Flow of Configuration Data

Message Prioritization

Msg. System

Adapter-spec. queues

**6.40 / 7.00 Message Processing
without prioritization**

- In previous releases we have adapter-specific queues. The number of maximum threads could be configured per direction (sender, receiver), server node and adapter type with the help of the Visual Administrator.

Message Prioritization

Dispatcher *new*

Msg. System

high

normal

low

Adapter-spec. queues

- **Outline: „Works as in ABAP IS"**

- **Weighted msg. Delivery for async. Messages**

- **Probability high / normal / low: 75:20:5**

- **Configuration in RWB:**

7.10 Message Processing with prioritization

Remarks:
- **Message Prioritization is designed to avoid that ‚prio high' messages get stuck due to a backlog of other messages on the same node**
- **To ‚see it working' at all you need a backlog in the messaging system**
- **To see the expected ratio in message delivery you need in addition comparable fast backends**

- Message Prioritization is designed to avoid that ‚prio high' messages get stuck due to a backlog of other messages on the same node.

- To ‚see it working' at all you need a backlog in the messaging system.

- To see the expected ratio in message delivery you need in addition comparable fast backends.

Message Prioritization in AAE – Configuration

- Rules within Component Monitoring with 3 different categories (high, low, normal) per Sender/Receiver Party/Component, Interface, Namespace

- Corresponding to properties „messaging.prioritization.high", „messaging.prioritization.low" and „messaging.prioritization.normal" of „XPI Service Messaging System" that can be set per server node (online modifiable)

- All QoS supported (EOIO serial per scenario)

- Prioritization during runtime takes place in principle whenever for a given adapter type and at a given moment of time all worker threads are occupied

- View Dispatcher Queue: RWB -> Component Monitoring -> Adapter Engine -> Status

---
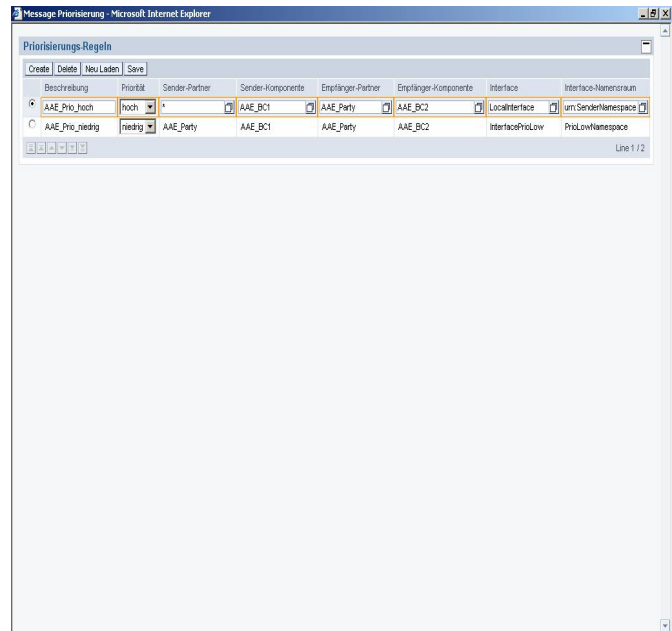
- Prioritization.dispatcherThreadCount: by default equal 1; measurements show no significant performance increase if this parameter is changed; one dispatcher per server node

- Maintain values in percent but this is just a rough estimation (i.e. percent of number of messages; if a message is processed, there is a certain probablilty that this is a high/medium/ low priority message in accordance with the configuration; default: 75/5/20%)

- Most specific configuration is used

- If nothing is configured for a certain scenario: priority medium is assigned

- If for one individual adapter type all the worker threads are occupied (at a given node), prioritization takes place

- Dispatcher Queue can be monitored: RWB => Component Monitoring => Adapter Engine => Status

Message Prioritization – Sender via IS
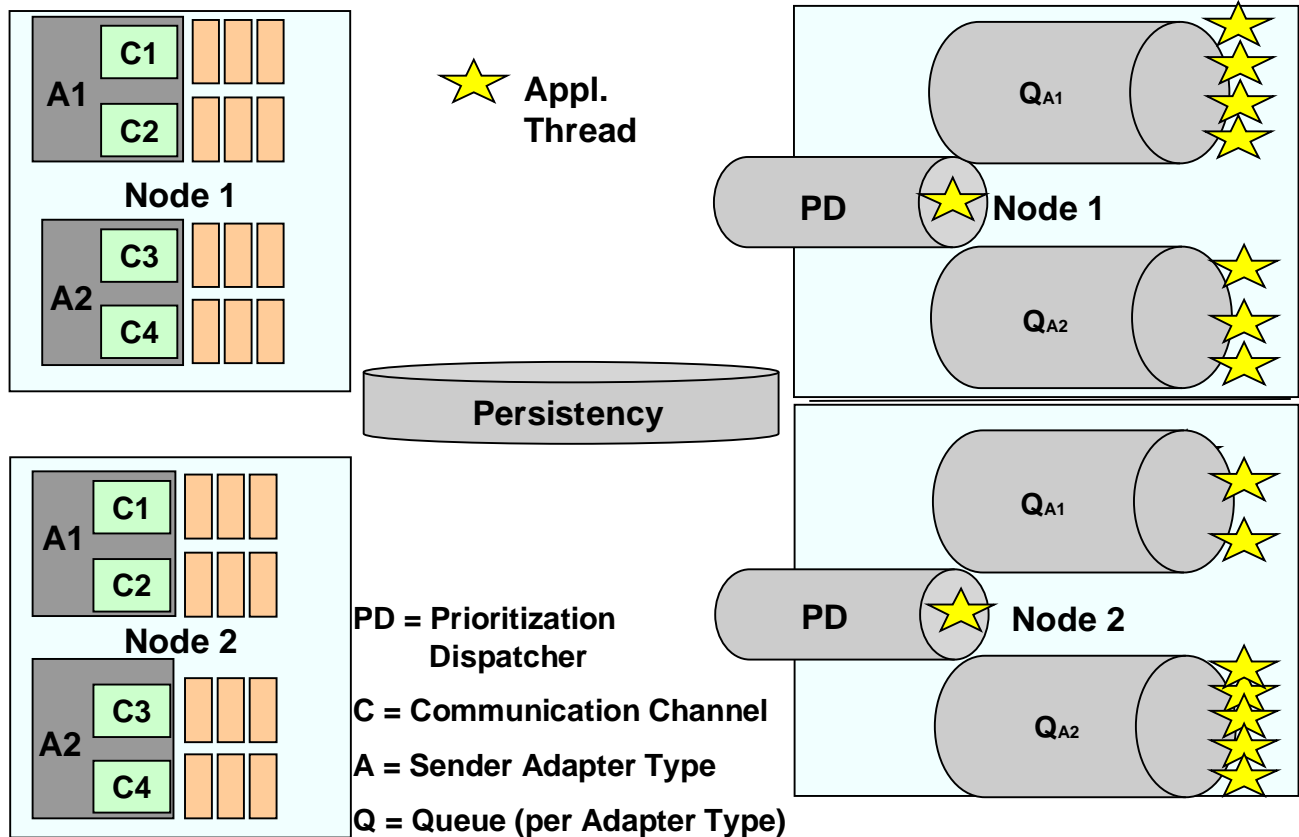
- The picture is valid for all QoS; to be more exact: 4 queues per adapter type (sync/async, sender/receiver); for all adapters even with own handler

- Performance depends also very much on the adapter's implementation (session/connection pooling, multithread etc.)

- Number of worker threads is configured per adapter type

- 7.0 Queue Separation: You can configure the max worker count values according to your specific needs, using the "messaging.connectionDefinition" property of the "SAP XI AF Core" service. Using the "name=global" entry some global template settings are defined. For each adapter type these defaults can be individually overwritten, by adding additional configuration entries, according to the following syntax:

  messaging.connectionDefinition=(name=global,...)(name=<AdapterTypeIdentifier>, Send.maxConsumers=<a>, Recv.maxConsumers=<b>, Call.maxConsumers=<c>, Rqst.maxConsumers=<d>)
  with <AdapterTypeIdentifier>=<AdapterType>_<Namespace>
  (e.g.: File_http://sap.com/xi/XI/System))
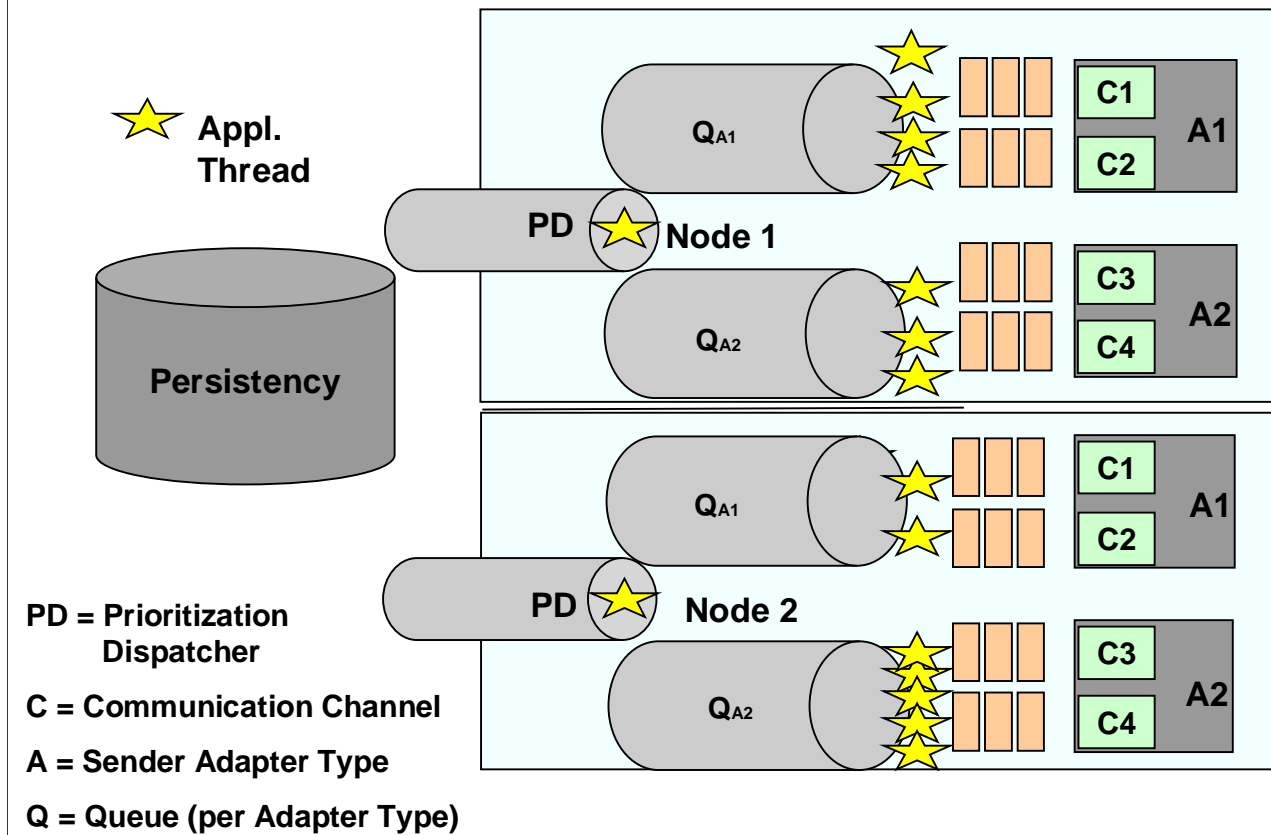
**Appl.
Thread**

Persistency

**PD = Prioritization
    Dispatcher**

C = Communication Channel

A = Sender Adapter Type

Q = Queue (per Adapter Type)

- The picture is valid for all QoS; to be more exact: 4 queues per adapter type (sync/async, sender/receiver)
- Number of worker threads is configured per adapter type!

Audit Log Cache

- 3 Modes: DB, Cache&DB (2000 entries per node), non

- Audit Log Memory Cache (stores audit log data in memory;
  in case of errors the log will be written to database)

- Configuration: Service „XPI Service: Messaging System"
  component name „com.sap.aii.af.ms.svc"

  Property : messaging.auditLog.memoryCache   default false

  Property: messaging.auditLogEnabled   default true

- Audit Log Memory Cache (stores audit log data in memory, if processing fails audit log will be written to database)
- Cache in memory: 2000 per node
- The services are „Online Modifiable"
- Especially for the AAE with local processing mode writing audit logs is relatively expensive

Q & A

Q1: Which adapters are supported?

A1:   - All technical adapters running in the Adapter Framework

- No iSpeak-Adapters (RNIF/CIDX), no IDoc/http-Adapter, XI-Adapter

- Third party upon release (no Adapters that require ccBPM)

Q2: Which mapping types are supported within the AAE?

A2:   - Message Mapping, Java, XSLT (no ABAP)

Q3: Now, since mapping and routing is carried out on the AAE, can we change the sequence of individual pipeline steps?

A3:   - No

Q4: Can the locally processed messages be seen within SXMB_MONI?

A4:   - No

Q5: Is there any change/case distinction in developing and testing adapters between a classic configuration via IS and integrated object configuration within the Advanced Adapter Engine?

A5:   No. Even there does not exist an official/released API that e.g. modules can detect if they run in a classical or Integrated Object mode.

- Plain-http can be implemented with the SOAP-Adapter; the XI adapter can be implemented technically on project basis with SOAP-Adapter in Axis-mode
- iSpeak: scenarios often use IDoc, ccBPM

On the Radar – Stay Tuned

- **Performance Monitoring**

- **Integrated Configuration Objects via Directory API**

- **Message persistence after mapping (optional)**

- **Content Based Routing**

- **Condition in interface determination**

- **Split of any kind (Split-Mapping, multiple receivers)**

- **Multimessage-Mapping/Enhanced Interface Determination**

- **Extended Receiver Determination**

- **IDoc-Adapter/WS-RM Adapter**

- **Improvements of cache mechanism for full cache refresh and mapping cache**

- **Adjustment of Sizing Guide/Quicksizer (planned for URS PI 7.1)**

Important Remark: Subject to change at any time without prior and further notice!!!

- Support of ICO within the Directory Configuration Assistant

- Flags „Maintain order at runtime" (Interface Determination)

- Message persistence after mapping (optional): SOX compliance

Agenda

1. Introduction and Architecture
2. Features and Configuration
3. **Integration Scenarios**
4. Summary

Scenario 1 – Performance and Resource Consumption

S1

S2

S3

IS Central

S1 => BPM/ABAP-services

S2/3 => intensive mapping steps
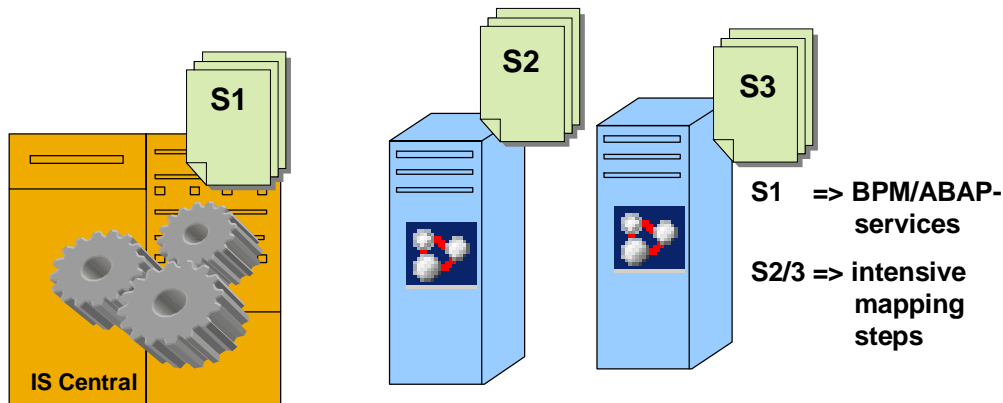
+ **Lower processing times, Speed (improved response times), less latency, increased throughput**

+ **Better support for synchronous scenarios**

+ **Flexibility in distributing pipeline resources (scenario outsourcing)**

+ **Dedicated AAE for scenarios with high throughput requirements**

S = Scenario

■ Even with central AAE there is less resource consumption and lower processing times in average for an individual message for a certain scenario compared to the same scenario with the „traditional configuration" via IS

Scenario 2 – Network Constraints

Need for decentral Adapter Engine and limited network capacities or firewall restrictions

S1

S2

IS Central

S1 => BPM/ABAP-services

S2/3 => intensive mapping steps

+ Less communication overhead

+ Large payloads are not transferred across different network domains
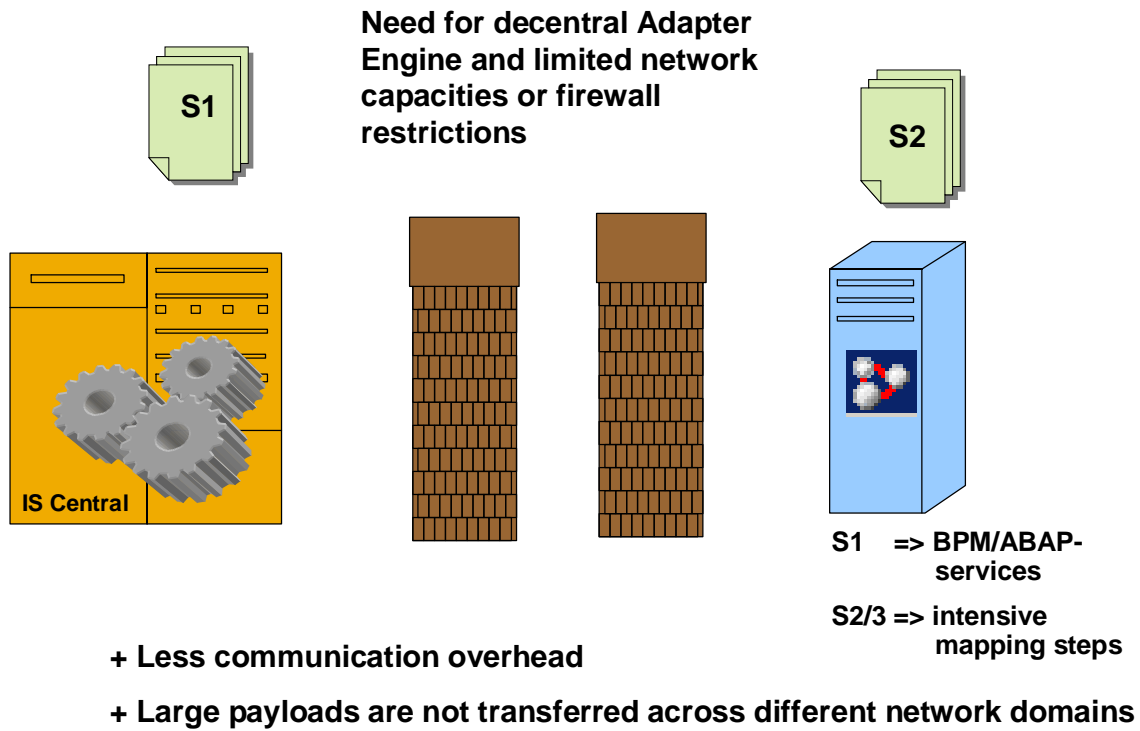
S = Scenario

Scenario 3 – Connect to IS, PCK, Proxy, AAE Natively

IS Central

S1 => BPM/ABAP-
services

S2/3 => intensive
mapping steps

S1

S1&
S2

S1&
S2
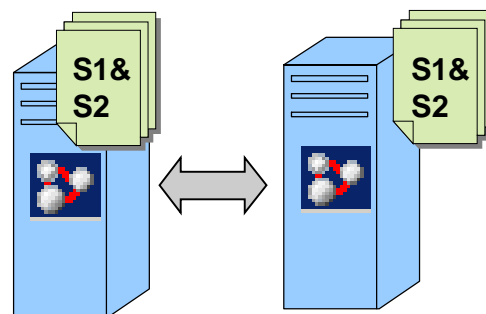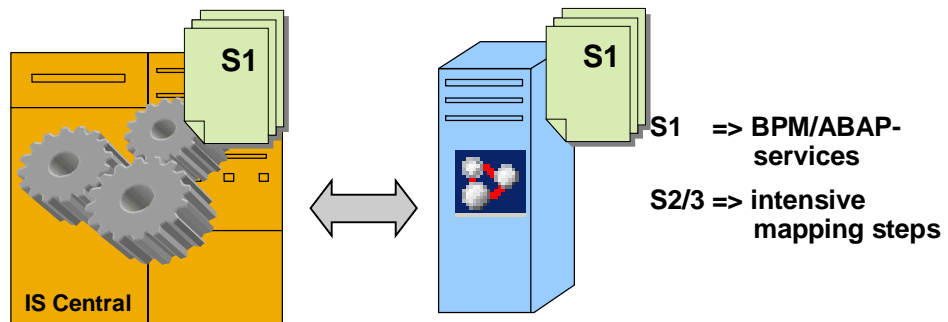
- No XI-Protocol or WS-RM

- No PCK-replacement

S = Scenario

Agenda

1. **Introduction and Architecture**
2. **Features and Configuration**
3. **Integration Scenarios**
4. **Summary**

AAE – Preliminary Results PI 7.1

Throughput - Soap2Soap
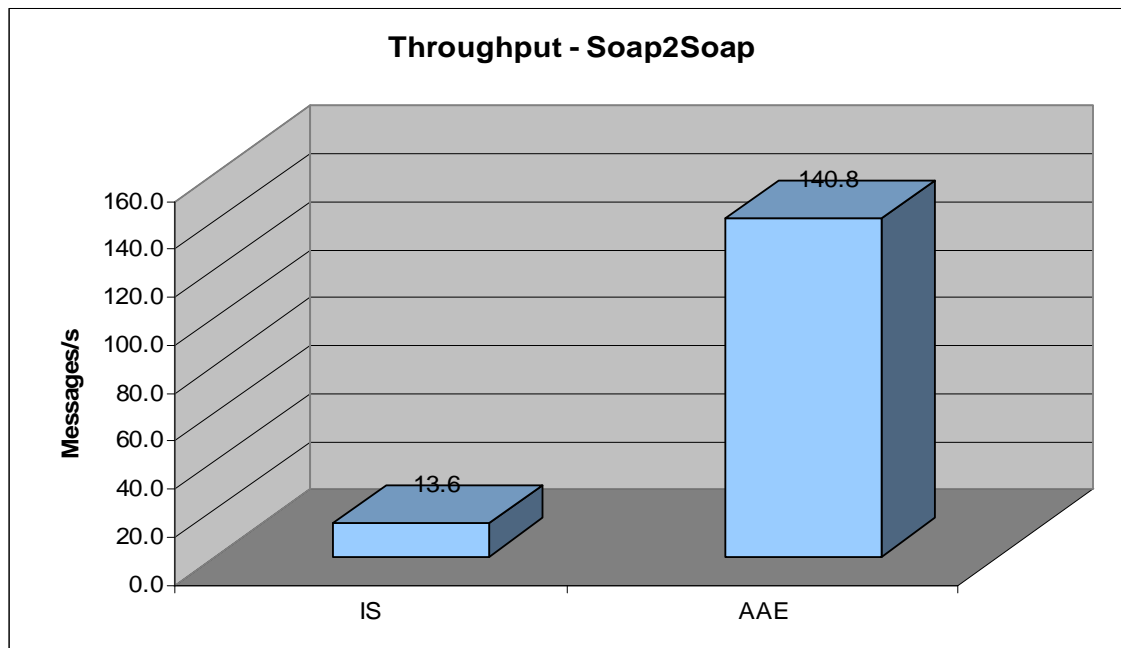
- System: 4 x Opteron 2.6 GHz
- Scenario without Mapping and without Packaging for IS;
  Mapping reduces comparison factor from 10 to 7-8
- Audit Log Memory Cache enabled for AAE processing

- Throughput: messages/s; 8Kb Message Size; boost message processing throughput up to 10 times; QoS=EO

- Significant performance improvements can be achieved always with local processing (if available) for a certain scenario;

- This is valid for all adapter types; huge mapping runtimes, slow networks, slow (receiver) applications reduce the throughput and therefore, rated for the overall scenario, also reduce the benefit of local processing in terms of overall throughput and response time measurement.

- The greatest benefit is recognized for small payloads (comparison: 10k, 50k, 500k) and asynchronous messages.

Synchronous Processing Response Times - Examples

Response Time Comparison - Soap2Soap
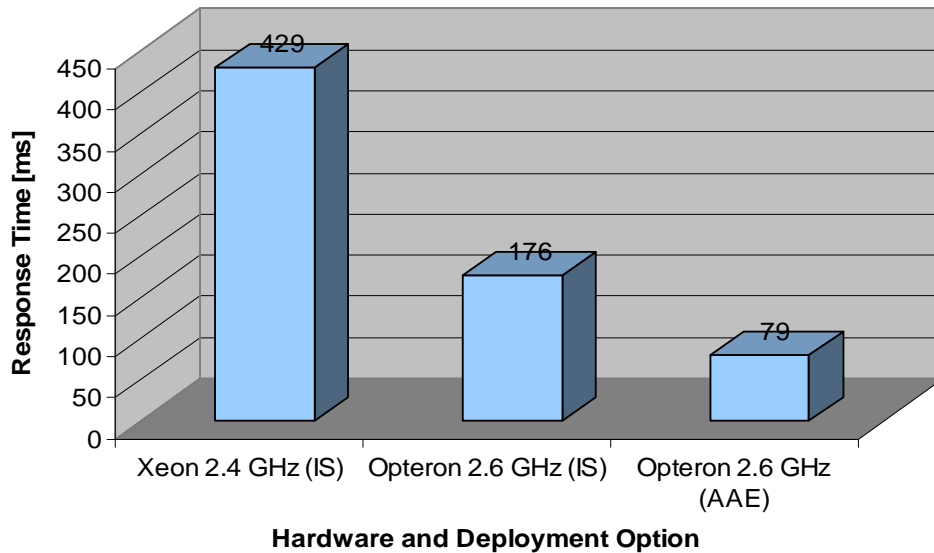
- Response times depend on backend service performance, additional used PI services (e.g. content-based routing, mapping) and underlying hardware
- Advanced Adapter Engine local processing can reduce response times to <100 ms

---

- Details about the sceanrio: end-to-end; QoS=BE with WebService Client⇔AAE⇔Provider(no application logic)

Summary

SAP

- **The Advanced Adapter Engine also allows for end-to-end message processing without invoking the central ABAP-based pipeline of the Integration Server**

- **If local processing is configured, several communication calls and persistency steps can be eliminated**

- **Reduces resource consumption and latency and increases message throughput**

- **Flexibility to deploy and run complete message broker scenarios on individual instances (decentral AAE); therefore improved scalability for large volumes by using autonomous processing units**

- The AAE is an additional feature required for an Enterprise Service Bus architecture.